Express Mail Label No. **EL089211025US** 01-10-00 #

UTILITY PATENT APPLICATION TRANSMITTAL (Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
12989 (JA998-189)

Total Pages in this Submission

jc710 U.S. PTO
01/07/00

TO THE ASSISTANT COMMISSIONER FOR PATENTS
Box Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

METHOD AND SYSTEM FOR SHARING THE BROWSER

jc511 U.S. PTO
09/480701
01/07/00

and invented by:

Makoto Kobayashi
Masahide Shinozaki
Takashi Sakairi

If a CONTINUATION APPLICATION, *check appropriate box and supply the requisite information:*

□ **Continuation** □ **Divisional** □ **Continuation-in-part (CIP)** of prior application No.: ____________

Which is a:

□ **Continuation** □ **Divisional** □ **Continuation-in-part (CIP)** of prior application No.: ____________

Which is a:

□ **Continuation** □ **Divisional** □ **Continuation-in-part (CIP)** of prior application No.: ____________

Enclosed are:

Application Elements

1. ☒ Filing fee as calculated and transmitted as described below

2. ☒ Specification having 23 pages and including the following:

   a. ☒ Descriptive Title of the Invention

   b. □ Cross References to Related Applications *(if applicable)*

   c. □ Statement Regarding Federally-sponsored Research/Development *(if applicable)*

   d. □ Reference to Microfiche Appendix *(if applicable)*

   e. ☒ Background of the Invention

   f. ☒ Brief Summary of the Invention

   g. ☒ Brief Description of the Drawings *(if drawings filed)*

   h. ☒ Detailed Description

   i. ☒ Claim(s) as Classified Below

   j. ☒ Abstract of the Disclosure

UTILITY PATENT APPLICATION TRANSMITTAL (Large Entity) *(Only for new nonprovisional applications under 37 CFR 1.53(b))*	Docket No. 12989 (JA998-189)
	Total Pages in this Submission

Application Elements (Continued)

3. ☒ Drawing(s) *(when necessary as prescribed by 35 USC 113)*

   a. ☒ Formal Number of Sheets 6

   b. ☐ Informal Number of Sheets ____

4. ☒ Oath or Declaration

   a. ☒ Newly executed *(original or copy)* ☐ Unexecuted

   b. ☐ Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional application only)*

   c. ☒ With Power of Attorney ☐ Without Power of Attorney

   d. ☐ *DELETION OF INVENTOR(S)*
   Signed statement attached deleting inventor(s) named in the prior application,
   see 37 C.F.R. 1.63(d)(2) and 1.33(b).

5. ☐ Incorporation By Reference *(usable if Box 4b is checked)*
   The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ☐ Computer Program in Microfiche *(Appendix)*

7. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all must be included)*

   a. ☐ Paper Copy

   b. ☐ Computer Readable Copy *(identical to computer copy)*

   c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

Accompanying Application Parts

8. ☒ Assignment Papers *(cover sheet & document(s))*

9. ☐ 37 CFR 3.73(B) Statement *(when there is an assignee)*

10. ☐ English Translation Document *(if applicable)*

11. ☐ Information Disclosure Statement/PTO-1449 ☐ Copies of IDS Citations

12. ☐ Preliminary Amendment

13. ☒ Acknowledgment postcard

14. ☒ Certificate of Mailing

    ☐ First Class ☒ Express Mail *(Specify Label No.)*: EL089211025US

UTILITY PATENT APPLICATION TRANSMITTAL **(Large Entity)** *(Only for new nonprovisional applications under 37 CFR 1.53(b))*	Docket No. **12989 (JA998-189)** Total Pages in this Submission

Accompanying Application Parts (Continued)

15. ☒ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

16. ☒ Additional Enclosures *(please identify below)*:

Associate Power of Attorney and Request for Change of Mailing Address

Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra		Rate	Fee
Total Claims	11	- 20 =	0	x	$18.00	$0.00
Indep. Claims	4	- 3 =	1	x	$78.00	$78.00
Multiple Dependent Claims (check if applicable) □						$0.00
					BASIC FEE	$690.00
OTHER FEE ***(specify purpose)***	Assignment recordal fee					$40.00
					TOTAL FILING FEE	$808.00

□ A check in the amount of to cover the filing fee is enclosed.

☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. **50-0510/IBM** as described below. A duplicate copy of this sheet is enclosed.

- ☒ Charge the amount of **$808.00** as filing fee.
- ☒ Credit any overpayment.
- ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
- □ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Signature

Richard L. Catania
Registration No. 32,608
SCULLY, SCOTT, MURPHY & PRESSER
400 Garden City Plaza
Garden City, NY 11530
(516) 742-4343

Dated: **January 7, 2000**

cc:

METHOD AND SYSTEM FOR SHARING THE BROWSER

Background of the Invention

Technical Field

The present invention relates to a method and a system for sharing a browser, particularly to a method and a system for implementing high-performance and real-time sharing of a browser for an existing Web browser and an existing Web page without having to install it by a user by embedding in the page itself a facility for controlling each element in the page.

Prior Art

Various methods are known for sharing a standard Web page in an as-is format by using an existing Web browser. (For example, see Japanese Patent Application No. Hei 9-254997). In such background art, however, it is necessary for every user to have a sharing system installed in advance. In Japanese Patent Application No. Hei 9-254997, synchronization of browser status is implemented by data exchange for sharing through the two interfaces, namely, an application interface (specifically, the functions of an information event of page loading, an inquiry about a current page and page setting) included in the browser and an interface at an operating system level (specifically, the functions of GET and SET for a message queue which a window of a browser has). At this time, the application interface and message queue interface are accessible only from outside a browser application process, and therefore, since a module for implementing the sharing is outside the browser, it must be installed in advance and should not be automatically downloadable as an applet. There was also a problem of having to install it on each individual platform since it relies on a browser-running OS or a window system.

Fig. 1 shows the working of browser sharing in Japanese Patent Application No. Hei 9-254997. To share a page of a Web server on a collaboration server, it is necessary that a program for sharing (WebShare) other than a browser is installed in advance for customers and a call center operator. This program for sharing allows sharing by hooking a browser API and an event. Also, such a program must be installed on each individual platform since it relies on a browser-running OS or a window system.

Summary of the Invention

The present invention is invented in view of the above-mentioned problem of the background art, and an object of the present invention is to provide a method and a system for sharing a browser wherein it is not necessary to install in advance a controller for sharing into a computer.

Another object is to provide a method and a system for sharing a browser which do not depend on the functions of an OS or a window system and operate on various platforms.

A further object of the present invention is to provide a method and a system for sharing a browser which are capable of an advanced sharing feature (for instance, allowing a customer to enter in a field of a form but prohibiting moving to another page) required in an application.

Still even further, an object of the present invention is to provide a method and a system for a real-time sharing a browser which require less data volume for sharing a browser and allow a good response even on a narrow band width as in an end user environment such as a home.

To attain the above objects, the present invention is organized as follows. First, a collaboration server is provided for accumulating pages to be shared from a Web server retaining original pages. A collaboration server (hereafter, merely a "server") comprises a CachinManager that accumulates pages dynamically generated on an original Web server for sharing, a CommunicationManager that controls sessions such as communication and participation/quitting among NodeManagers controlling a browser on each user machine, an ordinary Web server for downloading facilities for implementing sharing of the Web server (a NodeManager and a PageManager) and a Embedder that embeds in each page a PageManager for controlling pages.

The plural user machines to be shared comprise an existing Web browser capable of running Java and Script, and PageManager embedded in each page. A PageManager comprises two parts, namely, a PageController and a

PageCommunicator. A PageController detects changes in a page element, communicates them to another machine by way of a PageCommunicator, and receives changes in a page of another computer and then reflects the same changes to own page element. Changes in a page element refer to page loading, changes in value of text and buttons which are elements of a form, changes in a scroll position of a page, and operation of a remote pointer, etc. A PageCommunicator manages communication between a NodeManager and a PageController. A NodeManager for controlling a browser is provided for each process and manages communication between each PageManager and a server. Moreover, there is nothing unique for collaboration in the hardware configuration of the above-mentioned server. A user machine and a server are only different in name, and it is no problem if each user machine and server consist of exactly the same hardware.

09470140.091503

Fig. 2 shows a diagram of the entire configuration of the present invention. A collaboration server comprises a CachinManager that accumulates pages dynamically generated on an original Web server for sharing, an ordinary Web server (httpd) for downloading facilities for implementing sharing of a Web server, a CommunicationManager that controls sessions among NodeManagers on each user machine, and an Embedder that embeds a PageManager. The facilities for sharing plural computers (user machines) comprise two components, namely, a module for controlling each process of a browser (Web browser 1 or Web browser 2) (NodeManager) and a module for controlling each page (PageManager). A PageManager monitors a state of each page element in a page, detects changes and remotely exchanges information with a corresponding PageManager so as t dynamically perform setting of each page element to be in the same state. Also, for synchronization in a window of a nested frame structure, a PageManager checks a hierarchical structure of a frame (n-th position of n-th nest) and, with this as an ID, communicates with a corresponding PageManager. This hierarchical structure information can be obtained on any browser without being limited by a facility of cross frame security. While there are two user machines in Fig. 2, it is possible to share a browser likewise with a three or more machines.

A NodeManager controlling a browser performs communication (session and synchronization) between each PageManager and a server. A NodeManager resides in a page independent from the shared Web window and which does not migrate, and controls communication between PageManagers dynamically generated/terminated for each page loading. It also controls information across pages such as history. A PageManager and a NodeManager are embedded as Java applets which have an identical domain. Thus, regardless of the domain of the original page in which a PageManager is embedded, data communication by shared memory is performed between a PageManager and a NodeManager on any browser without being limited by a facility of cross frame security.

By configuring them in such a manner, sharing of a real-time Web browser becomes possible since a facility for page sharing can be embedded in an existing HTML page between an existing Web server and a browser without changing its original structure.

Brief Description of the Drawings

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings in which:
Fig. 1 is a diagram showing example of conventional browser sharing.

Fig. 2 is a diagram showing an entire configuration of the present invention.

Fig. 3 is a diagram showing operation inside a browser of the present invention.

Fig. 4 is a diagram showing example of processing of page loading in a nested frame.

Fig. 5 is a flowchart showing how it is until a browser can be shared the method of the present invention.

Fig. 6 is a flowchart for sharing user operation of a browser after the browser can be shared by the method of the present invention.

Fig. 7 is a diagram showing an example of hardware configuration of a server and plural computers used in the present invention.

Detailed Description of the Preferred Embodiments of the Invention

The operation of a browser sharing system is explained in detail below based on operation inside a browser in Fig.3.

1. Start of a NodeManager

A NodeManager is loaded into a new browser window as it is opened when a customer has logged in on a page which is a start of sharing or by pressing a start of sharing button. This window exists, if seen from a user, independently from a window to be shared and is not closed during a sharing session. In this window, a user interface for switching a remote pointer and a normal mode, etc. are displayed. A socket of Java is placed between this NodeManager and a CommunicationManager that is component of a server so that the shared data may be exchanged among plural user machines. In addition, shared memory is allocated as a class variable of a NodeManager applet. In this, a queue of a message with a PageManager is organized.

2. Embedding in a shared page of a PageManager A PageManager comprises PageController consisting a script for controlling each page element and a PageCommunicator made of an applet which communicates with a NodeManager through a message queue. These two modules are embedded by an embedder module of a server. As an example of actual code, the following code is embedded immediately after a <BODY> tag comes out while reading character strings in an HTML page.

```
<SCRIPT SCR="http://collabsvr/pagemgr.js"></SCRIPT>

<APPLET CODE="http://collavsvr/pagemgr.class"></APPLET>
```

As a PageCommunicator and a NodeManager have an identical domain, data can be exchanged avoiding a problem of security regardless of the domain of the page to be embedded. Thus, a multi frame page comprising plural domains can be shared.

3. Detecting changes in a page element in a shared page by a PageManager

A PageController detects local changes to the following elements and remotely communicates them to a corresponding PageController so as to implement synchronization of pages.

3-1. Mechanism for synchronizing page loading changes in a page occur along with the following operation.

(1) A click on a link in a page by user operation.

(2) Operation on a browser menu (selecting Back/Forward button and a bookmark) by user operation.

(3) Autoload by description in a page (designation by META syntax and description in an applet or a script).

Of these, (1) is synchronized, as in the detection of a changing event of a form element described below, by detecting and communicating a click event. In the cases of (2) and (3), since a PageController on a current page cannot detect a load request event, a PageController in a newly loaded page posts a new URL to another node. Here, Fig. 4 shows an example of processing page loading in a nested frame. Step 1 of Fig. 4 illustrates the initial state of browsers 1 and 2. Fig. 4 illustrates a case where the following initial state (synchronous state) switches to a final state (synchronous state) (a case where frame set 2 jumps to frame set 3).

```
Initial state
FRAMESET [S1]
   FRAME [F1]
   FRAMESET [S2]
       FRAME [F2]
       FRAME [F3]
       FRAME [F4]
```

```
Final state
FRAMESET [S1]
   FRAME [F1]
   FRAMESET [S3]
   FRAME [F5]
   FRAME [F6]
```

Meanwhile, S1 and S2 are abbreviation for frame set 1, frame set 2, and F1, F2, F3, F4, F5 and F6 are abbreviation for frames 1, 2, 3, 4, 5 and 6 respectively. Loading of a frame set is performed instead by a PageManager included in a page of the first frame (in the case that it is further nested and the first element is also a frame set, it should go below one by one).

In step 2 of Fig. 4, as frame set 2 jumps to frame set 3, PageManagers of frames 2, 3 and 4 are first destroyed and related message queues are released.

In step 3 f Fig. 4, frames 5 and 6 are loaded. Related message queues are placed, and a PageManager of frame 5 sends a message of "Load frame set 3."

In step 4 of Fig. 4, a NodeManager of browser 2 distributes a message to a message queue of an identical frame hierarchy, and a PageManager of frame 2 loads a URL of frame set 2. And it returns to a synchronous state.

3-2. Changing event of a form element

As for synchronization of form elements such as a text field or a button, changes are detected by the following two methods depending on their properties.

(1) Detecting a user-operated event with a hook of an event handler.

(2) As for an element which does not necessarily generate an event when changing, detecting an event through polling by a timer.

Explanation of the method of (2) is omitted since it is a background art. In the case of (1), since user operation to a form element such as a button generates a click event, it is possible to have so-called a hook of an event caused by a PageController setting its own handler to an onclick event handler. In general, however, there are causes where a user-defined event handler is already described in an element such as a button in JavaScript or VisualBasic Script, and in such cases, the original handler is not processed by merely replacing the event handler by a PageController so that the existing page does not operate as originally described. So, the original handler is rewritten on page loading to add a line for jumping to the handler for hooking. In this way, it becomes possible for a PageController to hook an event without affecting the original handler.

3-3. Page scrolling

While the position can easily be detecting by hooking an event generated on scrolling (onScroll), dragging scroll bar keeps generating events and causes too much traffic, so the position is detected through polling by a timer.

3-4. Operation of a remote pointer

A remote pointer may be added to an arbitrary page, in the case of Internet Explorer of Microsoft Corp. for instance, by adding it as a DIV element as follows. It can also be done with the same technique in the case of Netscape Communicator of Netscape Communications Corp.

```
var str = '<DIV style="position:absolute; overflow:none;
width=50px height=50px><IMG src="pointer.gif"></DIV>';
documet.body.insertADjacentHTML("AfterBegin", str);
```

A pointer is moved by moving DIV to a corresponding position with a mouseMove event to a window.

Fig. 5 explains on a flowchart the processing by the method of the present invention until a browser can be shared. When a user has logged in on a page which is a start of sharing or by pressing a start of sharing button provided on a page, a process of sharing is started. In step 510, a new browser window is opened, and a NodeManager is loaded there. This window of a NodeManager exists independently of a shared browser window, and is not closed during a sharing session. In this window of a NodeManager, a user interface for switching a remote pointer and a normal mode, etc. may be displayed. In step 520, a CommunicationManager and a NodeManager which are components of a collaboration server are connected by a socket of Java. This socket is used to exchange information for sharing among nodes. In step 530, a PageManager is embedded in an HTML page with an embedding module of a server. However, a PageManager comprises a PageController consisting of scripts for controlling each page element and a PageCommunicator consisting of applets for communicating with NodeManagers through a message queue. Since a PageCommunicator and a NodeManager are an identical domain, data can be exchanged avoiding a problem of security regardless of the domain of an HTML page to be embedded. Because of this, it becomes possible to share a multi frame page comprising pages of plural domains. The above concludes processing until sharing a browser becomes possible.

Fig. 6 explains on a flowchart the process for sharing browser operation of a user after browser sharing becomes possible by a method of the present invention. In reality, however, due to constraints of describing it on a flowchart, when an event is generated and a handler set for the event is to be activated, a decision part is used instead. Also, a timer is set for regular processing even in the case of a process not using a handler, a mere loop is used instead in the flowchart.

In step 610, a browser loads an HTML page in which a PageManager is embedded. Next, in step 620, shared memory is assigned as a class variable of a NodeManager's applet. A message queue with a PageManager is created in it. Also, a PageController sets its own handler to an onclick event handler.

Furthermore, the same process is performed for other events as required. In step 630, a decision on termination is made. Sharing is terminated if a sharing termination button is pressed. In step 640, a decision on page loading is made. page loading is decided by a current PageController detecting and event of a click on the case of a click in a page by a user. Loading generated by operation on a browser menu (back, forward, a bookmark) or description in a page (META syntax, an applet, a scrip), etc. is decided by a PageController in a newly loaded page referring to a location attribute of a browser window.

In step 642, a page loading process is performed. In the case of page loading by user operation on a browser menu or description in a page, a PageController in a newly loaded page posts a new URL to another node. In step 644, a page is unloaded. The browser unloads a current page along with page loading. The browser unloads a current page along with page loading. And in step 646, the PageManager is terminated. In the process of unloading the current page, the PageManager on this page is terminated. At this time, the message queue with a PageManager is eliminated and this shared memory is released.

In step 650, a decision on the form is made. As for synschronization of forms elements such as a text field or a button, it is decided by the two methods depending

09420847.102500

on their properties. A user-operated event is decided by hooking an event handler. Changes in an element which does not necessarily generate an event when changing are decided by regularly checking the value with a timer. In step 655, form processing is performed. If the user operation is of a kind to generate an event, the same process as the page loading process n step 642 is performed. As for changes in an element which does not necessarily generate an event when changing, the value of the changed form element is sent to an another node. Processing returns to step 630 after that. In step 660, a decision on scrolling is made. Scrolling operation is decided by regularly detecting the position with a timer. In step 65, a scrolling process is performed. The position of a new scroll is sent to another node. Processing returns to step 630 after that. In step 670, a decision on a pointer is made. It is decided by a mouseMove event to a window. In step 685, a pointer process is performed. The position of a new pointer is sent to another node. A pointer can be added to any HTML page as follows as a DIV element.

```
var str ='<DIVstyle="position:absolute;overflow:none;
width=50px height=50px><IMG src="pointer.gif"></DIV>';
document,body,insertAdjacentHTML("AfterBegin", str);
```

A pointer is moved by moving DIV to a position acquired from a mouseMove event. Processing returns to step 630 after that. In step 680, a decision on receiving is made. A PageCommunicator decides whether a message has been received from another node by checking a message queue. In step 685, a receiving process is performed.

Depending on the contents of a received message, an appropriate process is performed.

A pointer is moved by moving DIV to a position acquired from a mouseMove event. Processing returns to step 630 after that. In step 680, a decision on receiving is made. A PageCommunicator decides whether a message has been received from another node by checking a message queue. In step 685, a receiving message, an appropriate process is performed.
When a message instructing page loading by user operation on a browser menu or description in a page is received, the received URL is set to the location of the window and the same HTML page is loaded. When a message changing value of a form element is received, the form element is changed as instructed. When a message changing a position of a scroll is received, the position of the scroll is changed as instructed. When a message changing a position of a pointer is received, the position of the pointer is changed as instructed.

Fig. 7 shows an example of an embodiment of hardware configuration of a server and plural computers (user machines) used in the present invention. System 100 comprises central processing unit (CPU) 1 and memory 4. CPU 1 and memory 4 are connected via bus 2 with hard disk device 13 as an auxiliary storage (or drives for storage media such as CD-ROM 26 and DVD 32) via IDE controller 25. Likewise, CPU 1 memory 4 are connected via bus 2 with hard disk device 30 as an auxiliary storage (or drives for storage media such as MO 28, CD-ROM 29 and DVD 31)

09407583.101400

via SCSI controller 27. Floppy disk drive 20 is connected with bus 2 via floppy disk controller 19.

A floppy disk is inserted into floppy disk drive 20, and a code or data of a computer program for giving instructions to a CPU and so on in synergy with an operating system to implement the present invention can be recorded on this floppy disk, etc., hard disk device 13 (or a storage media such as MO, CD-Rom and DVD) and ROM 14, which is executed by being loaded to memory 4. This code of a computer program can also be compressed or divided into two or more so as to be recorded on two or more media.

System 100 can further have user interface hardware ad comprise pointing device (a mouse, a joystick, etc.) 7 or keyboard 6 for entry and display 12 for providing a user with visual data. It is also possible to connect with a printer via parallel port 16 or connect with a modem via serial port 15. This system 100 can be connected with a network via serial port 15 and a modem or communication adapter 18 (Ethernet or Token-ring card) etc. so as to communicate with other Web servers, other computers and so on. In addition, it is possible to connect a remote transmitter-receiver with serial port 15 or parallel port 16 so as to exchange data by means of an infrared ray or an electric wave.

Speaker 23 receives a speech signal which is D/A (digital/analog) converted by audio controller 21 via amplifier 22 and outputs it as speech. In addition, audio controller 21 makes it possible to A/D (analog/digital) convert speech data received from microphone 24 and capture into the system speech data outside the system.

Thus, it may be easily understood that the server and plural computers in the present invention is implementable by a communication terminal with a communication facility including an ordinary personal computer (PC) or a workstation, a notebook PC, a palmtop PC, a network computer, various household electrical appliances with a built-in computer such as a TV set, a game console with a communication facility, a telephone, a fax, a portable telephone a PHS, and electronic organizer or combination of these. However, these components are taken as examples, and it does not follow that all of them are required components of the present invention.

The present invention can implement a feature of advanced real-time sharing of a browser among a number of users which could not be implemented by any background art. In addition, synchronization of page loading, synchronization on input operations of form elements, synchronization of scrolling operations, synchronization of remote pointers and synchronization of annotations are possible, and a client machine only requires an existing browser comprising functions of Java and a script. And, since it does not require any external program or a module plug-in t be installed, a browser sharing system which does not burden a user with installation and requires little data traffic for synchronization is provided.

While the invention has been particularly shown and described with respect to preferred embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and details may be made therein without departing form the spirit and scope of the invention.

09480701.010700

Claims

Having thus described our invention, what we claim as new, and desire to secure by letters patent is:

1. A system having a server and plural computers for sharing a browser, each computer has browser for browser pages, a PageManger controlling said pages, and a NodeManager controlling said browser for making a communicating between said PageManager said server, wherein said PageManager comprises:

means for detecting changes in own page, and sending said changes to said NodeManager that sends said changes to said server; or

means for receiving changes in a page of another computer from said NodeManager, and reflecting said changes to own page.

2. The system according to claim 1 wherein said server comprises:a CachinManager that accumulates pages;a CommunicationManager that controls sessions among said plural computers; and an Embedder that embeds in each page PageManager for controlling pages.

3. The system according to claim 1 wherein said PageManager has a PageController and a PageCommunicator, said PageController comprises:detecting changes in a page element, and sending said changes to said NodeManager by way of said PageCommunicator; or receiving changes in a page of another computer from said NodeManager by way of said PageCommunicator and reflecting the received changes to own page element.

4. The system according to claim 3 wherein said changes in a page element are changes in page loading, changes in a form element including text and buttons, changes in a scroll position of a page or operation of a remote pointer.

5. The system according to claim 1 wherein said PageManager analyzes from hierarchical structure of a page and communicates with a corresponding PageManager based on this analysis result.

6. The system according to claim 1 wherein said NodeManager resides in a page independent from the page in the shared browser and which does not migrate and controls communication between PageManagers dynamically generated/terminated along with page loading, etc.

7. The system according to claim 1 wherein said NodeManager controls page information including transition history of a page.

8. The system according to claim 1 wherein said PageManager and said Nodemanager are embedded as Java applets which have an identical domain and data communication by shared memory is performed between said PageManager and said NodeManager.

9. A server for sharing a browser among plural computers, comprising:

means for receiving from a computer a signal for sharing said browser;

means for sending to a computer a NodeManager controlling said browser;means for receiving from a computer a request for viewing a page on said browser;

means for sending to a computer, according to said request for viewing a page, a request page in which a PageManager controlling pages is embedded;

means for receiving page change information sent by said PageManager via said NodeManager; and

means for sending said page change information to another computer.

10. A method for sharing a browser among plural computers, comprising the step of:

on activating said browser of a computer, loading a NodeManager on the computer from a server;

establishing communication between said server and said NodeManager;

said NodeManager assigning shared memory;

on page viewing on said browser, embedding a PageManager on a requested page on said server;

establishing communication between said NodeManager and said PageManager via said shared memory; and

sending changes in a page on page viewing to said NodeManager via said shared memory, or receiving changes in a page of another computer from said NodeManager via said shared memory and reflecting said changes to a next page.

11. A medium having a program for sharing a browser among plural computers, said program having said computers implement the functions of:

establishing communication with a server;

assigning shared memory;

on page viewing on said browser, issuing a page request to said server;

receiving from said server a page in which a PageManager controlling pages is embedded; and

sending to said server changes in a page received from said PagerManager via said shared memory, or receiving changes in a page of another computer from said server and sending said changes to said PageManager via said shared memory.

METHOD AND SYSTEM FOR SHARING THE BROWSER

ABSTRACT OF THE DISCLOSURE

The present invention provides a method and system for sharing a browser wherein it is not necessary to install a controller for sharing on a computer in advance. More particularly, a collaboration server is provided for accumulating pages to be shared from a Web server retaining original pages. A collaboration server comprises a CachinManager that accumulates pages dynamically generated on the original Web server for sharing, a CommunicationManager that controls sessions such as communication and participation/quitting among NodeManager controlling a browser on each user machine, an ordinary Web server for downloading facilities for implementing sharing of a Web server and an Embedder that embeds in each page a PageManager for controlling pages. The plural user machines to be shared comprise and existing Web browser capable of running Java and Script, and PageManager embedded in each page. A PageManager comprises two parts, namely, namely, a PageController and a PageCommunicator. A PageController detects changes in a page element, communicates them to another machine by way of a PageCommunicator, and receives changes in a page of another computer and then reflects the same changes to own page element. A PageCommunicator manages communication between a NodeManager and a PageController. A NodeManager cor controlling a browser is provided for each process and manages communication between each PageManager and a server.

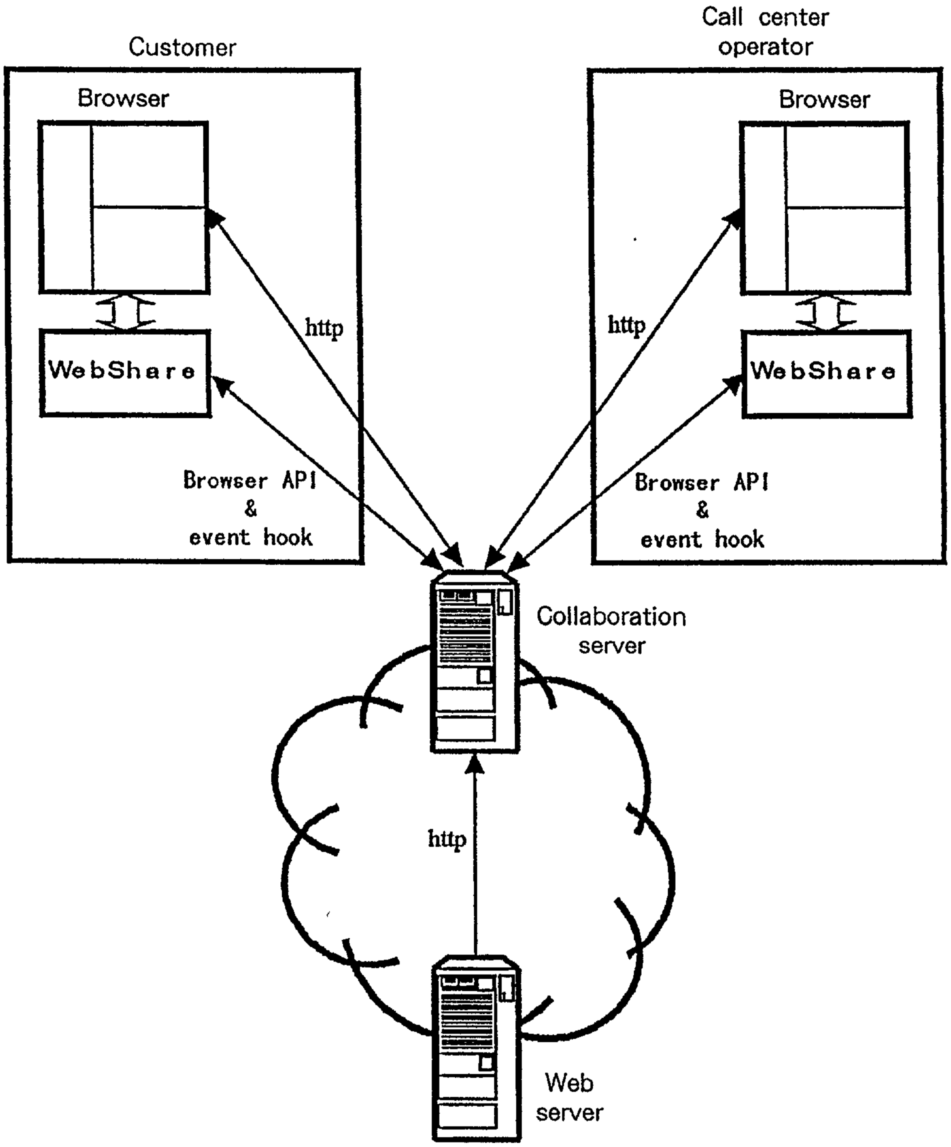

Fig. 1

09420703.070100

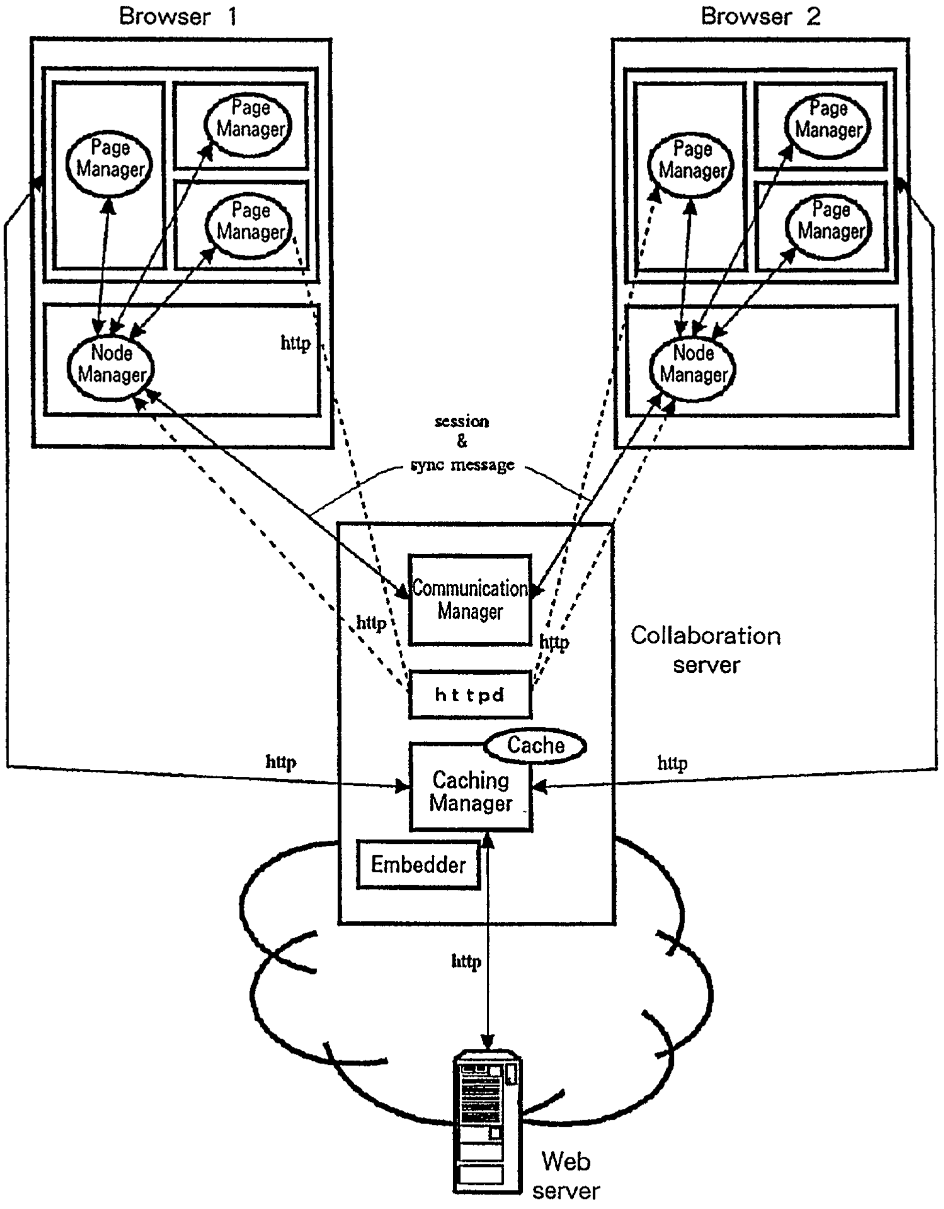

Fig. 2

09470102.01

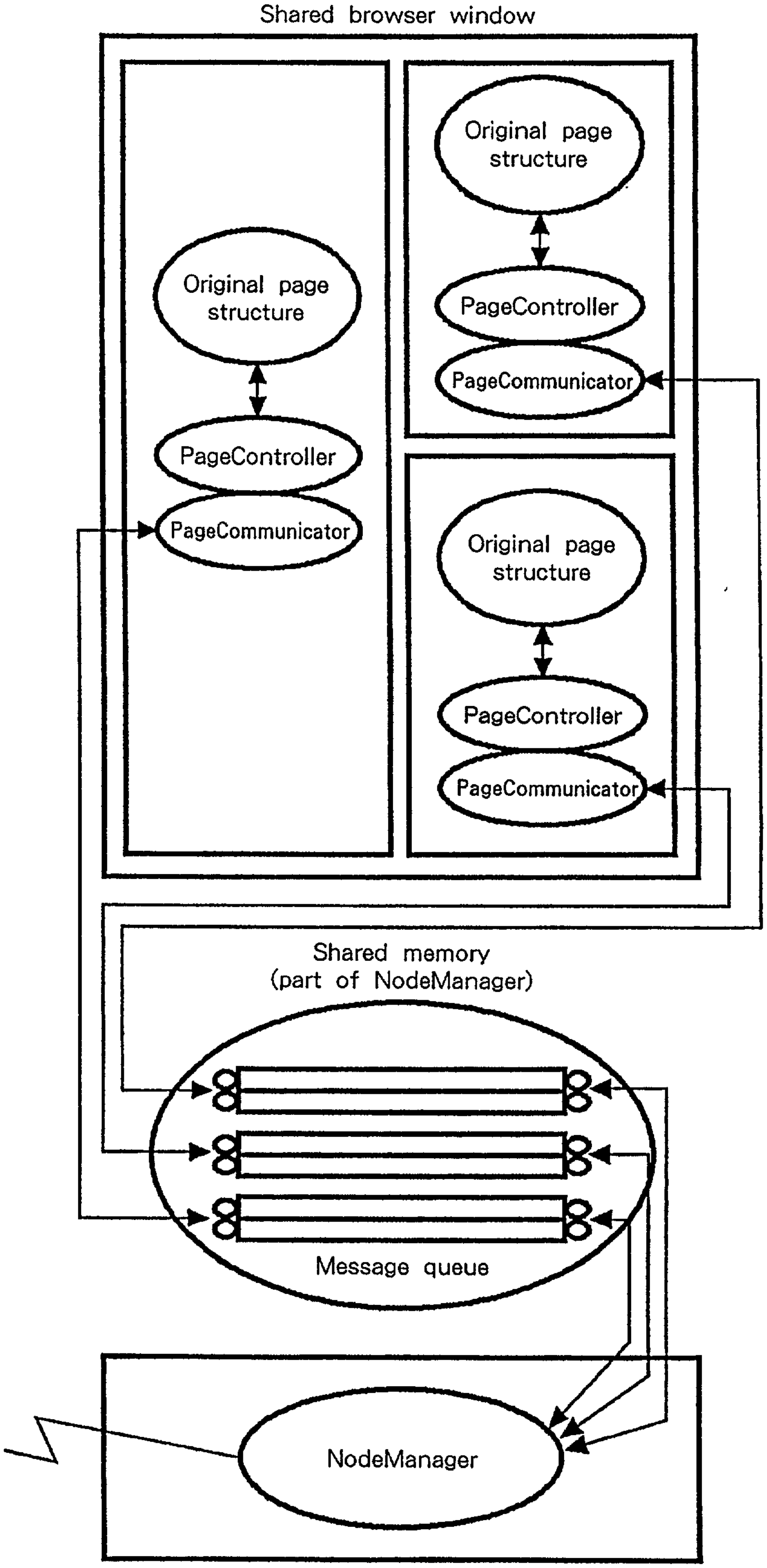

Fig. 3

09467204.012000

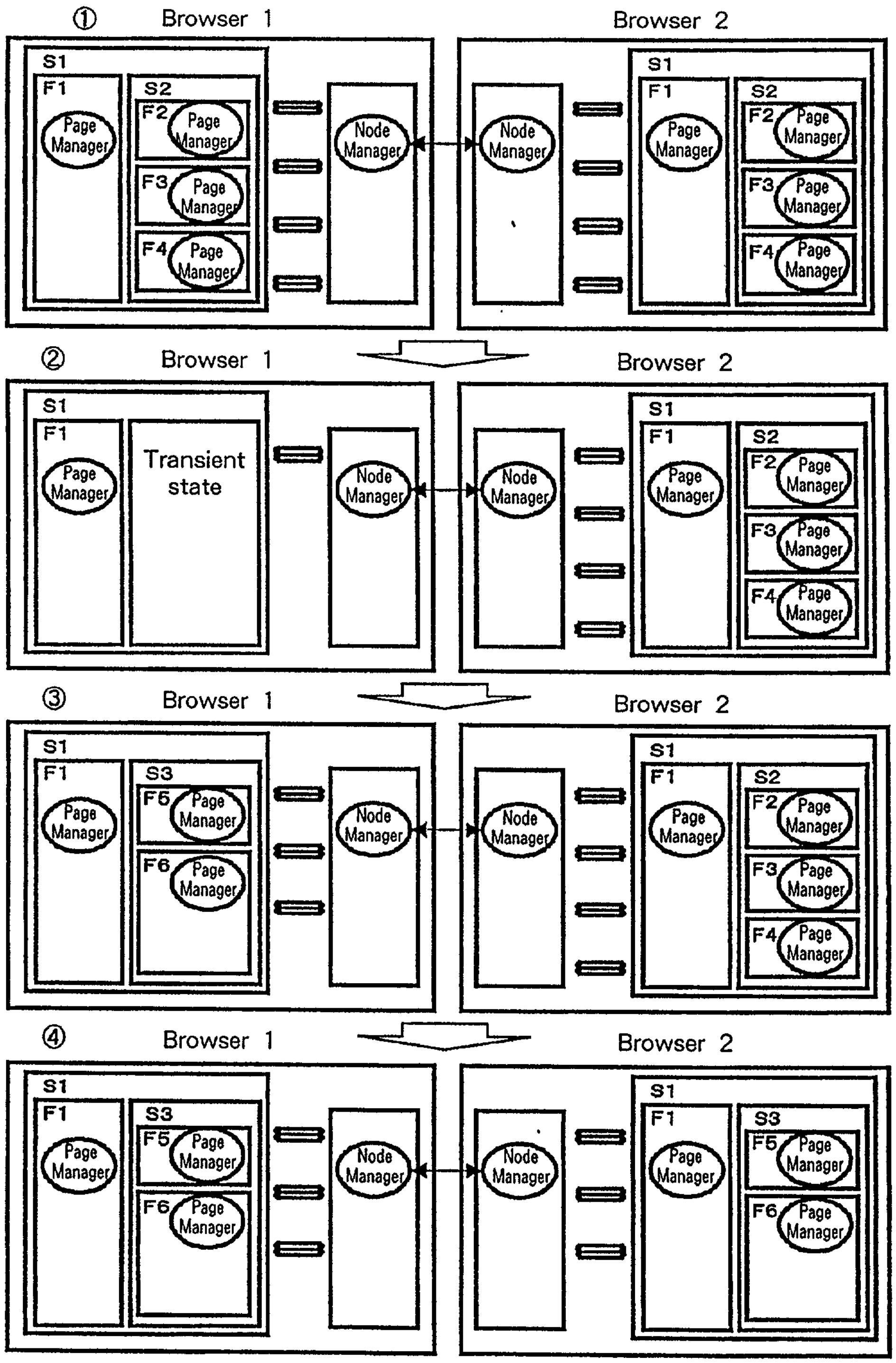
①
Browser 1
Browser 2
S1
F1
S2
F2
F3
F4
Page Manager
Node Manager
②
Transient state
③
S3
F5
F6
④

Fig. 4

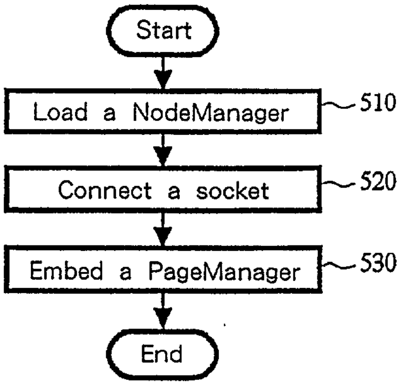
Start
Load a NodeManager
510
Connect a socket
520
Embed a PageManager
530
End

Fig. 5

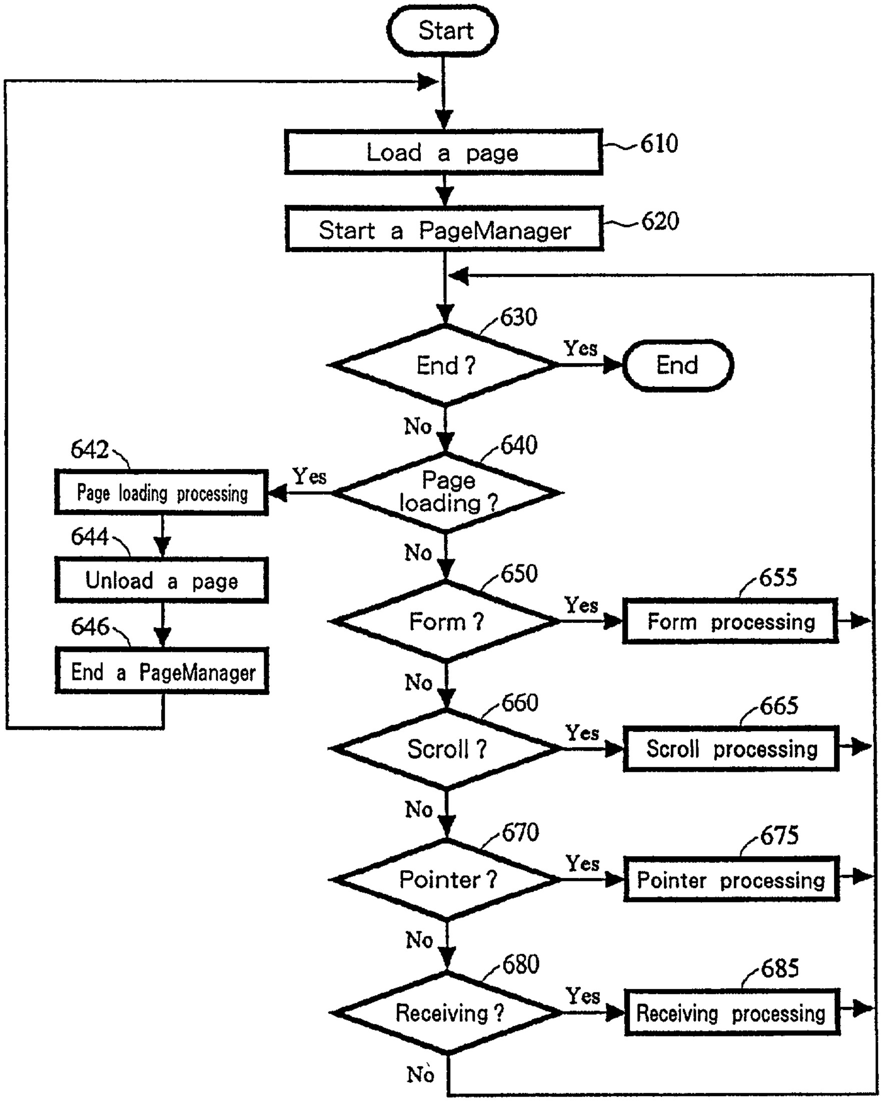
Start
Load a page
610
Start a PageManager
620
630
End ?
Yes
End
No
640
Page loading ?
Yes
642
Page loading processing
644
Unload a page
646
End a PageManager
No
650
Form ?
Yes
655
Form processing
No
660
Scroll ?
Yes
665
Scroll processing
No
670
Pointer ?
Yes
675
Pointer processing
No
680
Receiving ?
Yes
685
Receiving processing
No

Fig. 6

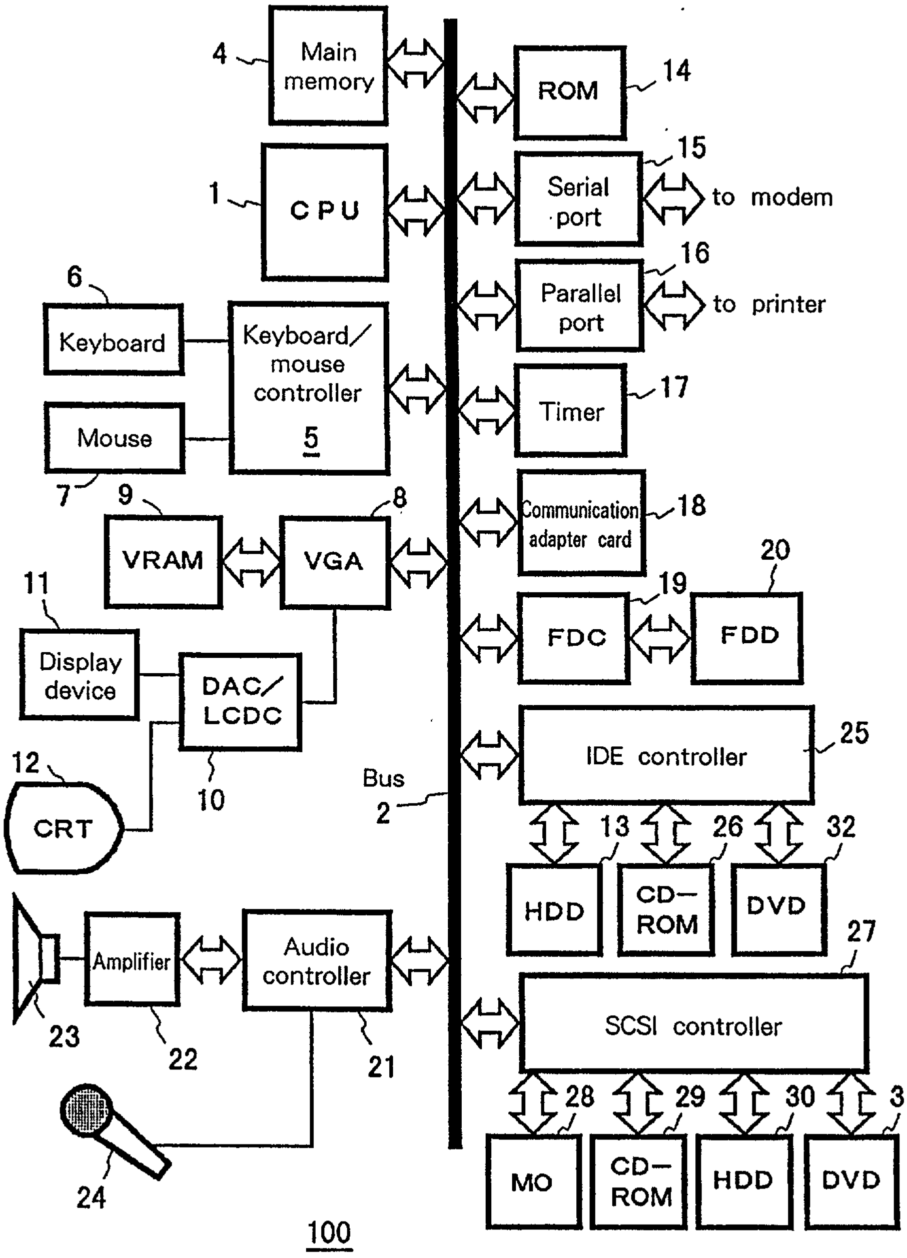
Main memory
4
CPU
1
Keyboard/ mouse controller
5
Keyboard
6
Mouse
7
VRAM
9
VGA
8
Display device
11
DAC/ LCDC
10
CRT
12
Bus
2
Amplifier
22
23
Audio controller
21
24
ROM
14
Serial port
15
to modem
Parallel port
16
to printer
Timer
17
Communication adapter card
18
FDC
19
FDD
20
IDE controller
25
13
26
32
HDD
CD− ROM
DVD
SCSI controller
27
28
29
30
31
MO
CD− ROM
HDD
DVD
100

Fig. 7

SSM&P Docket No.: 12989
IBM Docket No.:JA998-189

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: METHOD AND SYSTEM FOR SHARING THE BROWSER

the specification of which (check one)

_x_ is attached hereto.

___ was filed on ____________ as United States Application Number ____________

or PCT International Application Number ____________

and was amended on ____________ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or §365(b) of any foreign application(s) for patent or inventor's certificate, or §365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application, having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

			Priority Claimed	
11-013394 (Number)	Japan (Country)	21 January 1999 (Day/Month/Year Filed)	_x_ Yes	___ No
(Number)	(Country)	(Day/Month/Year Filed)	___ Yes	___ No
(Number)	(Country)	(Day/Month/Year Filed)	___ Yes	___ No

I hereby claim the benefit under 35 U.S.C. §119(e) of any United States provisional application(s) listed below.

(Application Number)	(Filing Date)
(Application Number)	(Filing Date)

I hereby claim the benefit under 35 U.S.C. §120 of any United States Application(s), or §365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States, or PCT International application in the manner provided by the first paragraph of 35 U.S.C. §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in 37 CFR §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith (list name and registration number).

Manny W. Schecter (Reg. 31,722), Terry J. Ilardi (Reg. 29,936), Christopher A. Hughes (Reg. 26,914), Edward A. Pennington (Reg. 32,588), John E. Hoel (Reg. 26,279), Joseph C. Redmond, Jr. (Reg. 18,753), Douglas W. Cameron (Reg. No. 31,596), Wayne L. Ellenbogen (Reg. No. 43,602), Stephen C. Kaufman (Reg. No. 29,551), Daniel P. Morris (Reg. No. 32,053), Louis J. Percello (Reg. No. 33,206), Jay P. Sbrollini (Reg. No. 36,266), David M. Shofi (Reg. No. 39,835), Robert M. Trepp (Reg. No. 25,933) and Louis P. Herzberg (Reg. No. 41,500).

SSM&P Docket No.: 12989
IBM Docket No.:JA998-189

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

Send Correspondence to: Richard L. Catania, Scully, Scott, Murphy & Presser
400 Garden City Plaza, Garden City, New York 11530

Direct Telephone Calls to: (name and telephone number) Richard L. Catania, (516) 742-4343

Makoto Kobayashi
Full name of sole or first inventor

Makoto Kobayashi Dec. 27, 1999
Inventor's Signature Date

1840-9 Kanamori, Machida-shi, Tokyo-to, Japan
Residence

Japan
Citizenship

Same as residence
Post Office Address

Masahide Shinozaki
Full name of second joint inventor, if any

Masahide Shinozaki Jan. 4, 2000
Inventor's signature Date

2-21-16-405 Tamagawadai, Setagaya-ku, Tokyo-to ,Japan
Residence

Japan
Citizenship

Same as residence
Post Office Address

Takashi Sakairi
Full name of third joint inventor, if any

Takashi Sakairi Dec. 27, 1999
Inventor's Signature Date

4266-2-804 Shimotsuruma, Yamato-shi, Kanagawa-ken, Japan
Residence

Japan
Citizenship

Same as residence
Post Office Address

09476370.010600

jc511 U.S. PTO
09/480701
01/07/00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Makoto Kobayashi et al. **Docket:** 12989

1999 DEC 21 A
RECEIVED

Serial No.: Unassigned **Dated:**

Filed: Herewith

For: METHOD AND SYSTEM FOR SHARING THE BROWSER

Assistant Commissioner for Patents
Washington, DC 20231

ASSOCIATE POWER OF ATTORNEY AND REQUEST FOR CHANGE OF MAILING ADDRESS

Sir:

Applicants, by their attorney of record, hereby grant and Associate Power of Attorney to:

RICHARD L. CATANIA, Reg. No. 32,608; FRANK S. DIGIGLIO, Reg. 31,346; KENNETH L. KING, Reg. No. 24,233; STEPHEN D. MURPHY, Reg. No. 22,002; LEOPOLD PRESSER, Reg. No. 19,827; JOHN S. SENSNY, Reg. No. 28,757; and EDWARD W. GROLZ, Reg. No. 33,705

with full power of substitution to prosecute this application and transact all business in the United States Patent and Trademark Office in connection therewith.

Applicants further request that all future correspondence in connection with this application be directed and addressed to:

RICHARD L. CATANIA, ESQ.
SCULLY, SCOTT, MURPHY & PRESSER
400 Garden City Plaza
Garden City, New York 11530

Direct all telephone call to: (516) 742-4343.

Respectfully submitted,

Wayne L. Ellenbogen

Wayne Ellenbogen
Registration No.: 43,602
Telephone No.: (914) 945-2587

IBM Corporation
T.J. Watson Research Center
Route 134/Kitchawan Road
P.O. Box 218
Yorktown Heights, New York 10598

09480701.010700